REMARKS

Claims 1-5, 7-11, 13-16, and 21-32 are pending. Claims 1-5, 7-11, 13-16 and 21-32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,363,378 to Conklin, in view of U.S. Patent No. 5,390,281 to Luciw, U.S. Patent 6,078,953 to Vaid, and U.S. Patent No. 6,513,031 B1 to Fries et al. Claims 1-5, 7-11, 13-16 and 21-32 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting over U.S. Patent Application No. 09/653,713 in view of Conklin, Luciw, Vaid, and Fries. Claims 1-5, 7-11, 13-16 and 21-32 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting over U.S. Patent Application No. 09/512,963 in view of Conklin, Luciw, Vaid, and Fries.

Reconsideration is requested. No new matter is added. The rejections are traversed. Claims 1, 7, and 13 have been amended. Claims 1-5, 7-11, 13-16, and 21-32 remain in the case for consideration.

REJECTION OF CLAIMS UNDER 35 U.S.C. § 103(a)

Rejection over Conklin, in view of Luciw, Vaid, and Fries

Referring to claim 1, the invention is directed toward a computer-implemented method for enforcing policy over a computer network, the method comprising: selecting a dictionary, the dictionary including a plurality of concepts organized as a directed set, exactly one concept identified as a maximal element, and a plurality of chains connecting the maximal element to other concepts in the directed set; selecting a set of chains to form a basis; selecting at least one concept in the dictionary; creating a state vector in a topological vector space for each of the selected concepts, wherein each state vector includes at least one measure of how concretely the concept is represented in each chain in the basis; assembling a first subset of the state vectors in the topological vector space into a template, the topological vector space including at least one vector not in the template; assigning a policy to the computer network; monitoring a content stream on the computer network to construct an impact summary including a second subset of the vectors in the topological vector space; and enforcing the policy when the impact summary is within a threshold distance of the template.

Referring to claim 7, the invention is directed toward a computer-readable medium containing a program operable on a computer to enforce policy over a computer network, the program comprising: selection software to select a dictionary, the dictionary including a plurality of concepts organized as a directed set, exactly one concept identified as a maximal

element, and a plurality of chains connecting the maximal element to other concepts in the directed set; selection software to select a set of chains to form a basis; selection software to select at least one concept in the dictionary; creation software to create a state vector in a topological vector space for each of the selected concepts, wherein each state vector includes as its components measures of how concretely the concept is represented in each chain in the basis; definition software to define a template, the template including a first subset of vectors in the topological vector space, the topological vector space including at least one vector not in the template; assignment software to assign a policy to the computer network; monitoring software to monitor a content stream on the computer network to construct an impact summary including a second subset of the vectors in the topological vector space; and enforcement software to enforce the policy when the impact summary is within a threshold distance of the template.

Referring to claim 13, the invention is directed toward an apparatus for enforcing policy over a computer network, the apparatus comprising: a computer; a directed set stored in the computer including a plurality of concepts, exactly one concept identified as a maximal element, and a plurality of chains stored extending from the maximal element to other concepts in the directed set; a basis including a subset of the plurality of chains; for at least one concept in the directed set, a state vector in a topological vector space, wherein each state vector includes at least one measure of how concretely the concept is represented in each chain in the basis; a template stored in the computer, the template including a first subset of the state vectors in the topological vector space, the topological vector space including at least one vector not in the template; a policy associated with the template; a monitor installed in the computer adapted to monitor a content stream in the computer network to construct an impact summary including a second subset of the state vectors in the topological vector space; and a policy enforcer adapted to enforce the policy when the monitor determines the impact summary to be within a threshold distance of the template.

Conklin

There are several differences between the invention as recited in 1, 7, and 13 and Conklin. First is that the claimed invention uses a dictionary organized as a directed set. As explained on page 4, line 30 to page 5, line 3 of U.S. Patent Application Serial No. 09/643,713, titled "INTENTIONAL-STANCE CHARACTERIZATION OF A GENERAL CONTENT STREAM OR REPOSITORY," (Intentional Stance application) which has been incorporated by reference, a directed set is a different concept than a tree. Among other

(mathematically equivalent) definitions, a tree is a set of nodes connected by edges, the tree having no self-loops and such that between any two nodes in the tree there is *exactly one* path between the two nodes. (See, e.g., Shimon Even, GRAPH ALGORITHMS 22 (1979), a copy of which is attached.). A directed set has no such limitation. Rather, a directed set is a set of nodes connected by edges, where there can be any number of distinct paths between the maximal element and any other element in the directed set. Indeed, as shown in the FIG. 2 of the Intentional Stance application, there are two different paths between "set" and "relation." One path goes through "product," the other path goes through "subset." Since a tree cannot have multiple paths between a pair of nodes, the trees of Conklin do not anticipate the directed set of the instant invention.

A directed set is also different from a directed graph because a directed graph does not require there to be one maximal element. By providing that elements in one ontology tree can cross-reference to another ontology tree, Conklin can be said to be teaching the use of a directed graph. A directed graph allows for there to be more than one directed path connecting two nodes. While it is possible for a directed graph to have one root node, it is not a requirement. A directed graph has a root *r* if, for every node *v* that is a vertex in the graph, *v* is reachable from *r*; i.e., there is a directed path which starts in *r* and ends in *v*. (See, e.g., Shimon Even, GRAPH ALGORITHMS 30 (1979), a copy of which is attached.)

Conklin's example in FIG. 6 shows no root node, or maximal element, in the graph. At best, each individual tree has a maximal element. For example, "France" and "art galleries and museums" are cross referenced elements linking the "geography" ontology with the "leisure and recreation" ontology. While this shows that there can be more than one unique path between two nodes, it still fails to show a maximal element, from which paths (or chains) lead to all other elements. Suppose for example that the root node is "geography". For this to be true there would need to be a directed path from "geography" to every other node. One can quickly see that many nodes are connected to "geography". In fact, even "places of interest" is connected to "geography", because "France" is cross linked to "places of interest". However, not every node shown in FIG. 6 is connected to "geography". For example, there is no directed path from "geography to "tourism". Therefore, "geography" cannot be a root node for Conklin's directed graph. Using the same logic, one can see that "leisure and recreation" also cannot be a root node, nor can any other node be a root node.

While Conklin shows that some elements can be tied together, he does not require a root node. Nor does Conklin provide examples suggesting that one element could be a "maximal element" of all other concepts in the knowledge base. Simple cross-referencing

between the two different trees does not make one element a maximal element of every concept in every tree.

In arguing that Conklin teaches a concept identified as a maximal element, the Examiner cites to column 7, lines 39-50 of Conklin, where two independent ontologies are described. Each of the ontologies that are shown in FIG. 3, have a maximal element. For in ontology 220 the root node is $node_A$, where ontology 230 has $node_B$ as its root. While this suggests a hierarchical language system, it also illustrates that Conklin does not teach a system with one maximal element. (FIG. 3 shows two independent roots, but no single maximal element.)

By using separate and independent trees for different ontologies, Conklin teaches a system wherein some elements cannot be compared. For example, referring to FIG. 6 of Conklin, there is no way to compare "Western Europe" with "tourism", as they are in different ontologies. In other words, there is no common reference point by which concepts in Conklin can be compared. In contrast, in the directed set of the instant invention, every pair of concepts has at least one common ancestor: the maximal element. Thus, at worst, every pair of elements is related through the maximal element. This makes it possible to compare disparate concepts such as "iguana" and "man."

Conklin also fails to teach or suggest chains connecting the maximal element to other concepts in the dictionary. The Examiner argues that Conklin teaches the establishing of chains, in that there is a path between any two nodes in a tree. But Conklin makes no mention of any significance to chains, and does not even use the term. And for this to be true there is the additional assumption that there could only be one ontology in Conklin, as with multiple ontologies, there cannot be a single maximal element as claimed.

In addition, such an interpretation of the meaning of the term "chain" ignores certain basic facts about chains as claimed. Chains are sets of concepts, connected by directed links, and start at the maximal element. That means that the concepts in chains move from the general to the specific, narrowing the focus of the interpretation, where directed links define "is a" relationships between pairs of concepts. Conklin does not teach or suggest this feature. And to analogize a chain to a path between nodes in a tree ignores stated features of the claims.

The Examiner argues that in FIG. 6, and column 12, lines 1-18 of Conklin, places of interest is a subcategory under the category, "tourism." Thus allowing "is a" to be a basis where each chain in the tree has a "is a" relationship to its parent. But this still does not

suggest that every concept in the dictionary has an "is a" relationship with the maximal element, only to the root element of the particular ontology that the concept resides. Thus, Conklin does not teach or suggest chains connecting the maximal element to other concepts in the dictionary.

Even if Conklin could be said to teach establishing chains, Conklin teaches nothing about selecting chains to form a basis. In mathematics, a basis is a set of vectors or other objects that span a subspace. While it might not appear that the chains that make up the basis span a subspace, in fact they do. Conklin has no analog to the concept of selecting chains to form a basis.

In addition, the concept of a "basis", aside from being discussed on page 8, lines 12-19 in the specification of U.S. Patent Application Serial No. 09/512,963, titled "CONSTRUCTION, MANIPULATION, AND COMPARISON OF A MULTI-DIMENSIONAL SEMANTIC SPACE," (the Construction application), is a well-defined concept. (*See, e.g.*, STEWART VENIT & WAYNE BISHOP, ELEMENTARY LINEAR ALGEBRA 146 (2d ed. 1985), a copy of which is attached.) Had Conklin intended to teach forming a basis for a subspace, he would have used the term. That he did not use the term "basis" anywhere in his patent makes clear that he does not view his "chains" (undefined as they are) as forming a subspace.

Furthermore, Conklin does not teach state vectors created in a topological vector space for each of the selected concepts. In arguing that Conklin teaches this feature, the Examiner cited to the document theme vector of Conklin. As shown in Table 1 in column 4 of Conklin, the document theme vector includes document themes, their relative strengths, and potentially categories for the themes. The Examiner did not cite anything in Conklin discussing measuring distances between vectors.

The claimed state vector, however, is not a document theme vector. A state vector represents concepts, which are independent of documents. It is worth noting that none of the claims uses the term "document". But the independent claims do describe "a state vector for each concept". Thus, the state vectors are not document theme vectors, but rather representations of the concepts in a Euclidean space. Thus, the only similarity between state vectors and Conklin's document theme vector is the use of the word "vector"; otherwise, the ideas are completely different.

Finally, the instant invention teaches measuring how closely each concept is represented in the basis chains. As discussed above, a chain is a set of concepts connected by directed links and originating at the maximal element, moving from the general to the specific. Thus, measuring how concretely a concept is represented in a basis chain involves measuring a distance between a node in the directed set and a set of concepts (i.e., more than one concept), and is not simply measuring a distance between two nodes. Conklin teaches measuring the distance between two specific nodes: a focal node and a query feedback node. As the objects being compared are quite different in the instant invention as compared with Conklin, Conklin cannot be said to teach or suggest the measurement taught in the instant invention.

The Examiner points to figure 4 and column 7, line 62 through column 9, line 26 of Conklin as teaching measuring how concretely a concept is represented in each chain in the basis. Here Conklin teaching using a weighting approach. But this reading of Conklin fails for two reasons. First, selecting chains for the basis has nothing to do with measuring the distance between the basis chains and concepts in the directed set, to which weighting would be applicable. Second, Conklin's weighting approach is highly dependent on the terms being searched in the trees, whereas the basis chains in the instant invention are selected without regard to any particular terms that might be searched. Thus, assigning weights to individual nodes in the Conklin ontologies cannot be considered to teach selecting chains to form a basis.

Therefore, there are several features in the independent claims that are not taught or suggested by Conklin. Conklin does not teach a dictionary organized as a directed set with one maximal element, or chains connecting the maximal element to other concepts in the dictionary. Conklin does not teach selecting chains to form a basis, and he does not teach state vectors created in a topological vector space for each of the selected concepts. Lastly, Conklin does not teach or suggest measuring how closely each concept is represented in the basis chains.

Luciw

The Examiner acknowledges that Conklin does not teach assembling a subset of state vectors into a template or associating an action with the template. The Examiner refers to Luciw for teaching these features. Luciw teaches a means of generating information based on a familiar series of computer events. It observes and interprets user and system behavior and then guesses what should be done based on that observation. However, instead of using a

directed set to generate topological state vectors, Luciw uses a frame-based approach and look-up tables.

As Luciw does not teach a dictionary comprised of a directed set with one element as a maximal element, Luciw also does not teach the use of templates with state vectors. In fact, no reference teaches any sets of vectors that could possibly provide a subset of state vectors. As such, Luciw cannot be said to teach the assembling of state vectors into a template.

The Examiner argues that Luciw teaches "assembling information" into the template, and that this makes obvious assembling state vectors into a template. But for this reasoning to have any possible justification, the concept of state vectors must be taught in one of the cited references. As argued above, Conklin fails to teach or suggest state vectors as claimed; thus, Luciw would have to teach or suggest state vectors. And Luciw's use of his template, as shown in FIG. 4a, uses the template as a form. Forms are not vectors, and Luciw makes no suggestion that the forms can be represented as vectors. In other words, Luciw does nothing more than use a word ("template") in common with this application; even if the general idea behind the word is quasi-similar, the actual use is quite distinct. Thus, Luciw's "template" does not include state vectors, and Luciw does not teach or suggest state vectors anywhere.

Vaid

Vaid teaches a system and method for monitoring quality of service in a network. Vaid uses a traffic management tool coupled to a firewall server. The traffic management tool includes a flow control module and a queueing control module. A bandwidth management tool classifies an information flow into portions, which are directed to the flow control module and the queueing control module.

The Examiner notes that Vaid does not teach a template with a subset of vectors in a topological vector space, monitoring a content stream to construct an impact summary of a second set of vectors in the topological vector space, for determining when the impact summary is within a threshold distance of the template.

Furthermore, Vaid does not teach a dictionary with one concept as a maximal concept in a directed set, and selecting a set of chains to form a basis. The only relevance Vaid has to the present application is that of teaching a network policy for traffic management. While this might speak to the use of a policy in the present application, it does not address using a policy associated with analysis of vectors.

Fries

Fries teaches a system from improving search area selection. Fries operates by parsing a natural language search query into keywords, and determining what specific categories of information (termed "goals" by Fries) the user is interested in. In FIG. 13, Fries shows a flowchart for getting topics from keywords. After keywords have been parsed from a search query, a topics list is obtained from the topics dictionary. The topics are calculated based on the natural language parse (NLP) bits. Then the topic hits are combined and sorted, and finally returned.

Operation of the topics dictionary is shown in a flowchart in FIGs. 14A-14B. The topics dictionary serves to return topics from a parsed sub-query. The topics dictionary is a list of topics that a user might be interested in based on the keywords listed in a search query. Nowhere does Fries describe the topics dictionary as a directed set. Instead, as shown in FIG. 14C, the topics dictionary has topics that are indexed and stored in a database.

In FIG. 17, Fries shows a flowchart of the training and use of the support vector machine (SVM). First, corpus queries are analyzed to identify corpus goals. Then corpus clue features are generated. Features can include semantic bits, topics, and user profiles. The corpus clue features and corpus goals are submitted to the SVM, which then identifies distances to the closed goal vectors in the vector space.

Just as Conklin, Luciw and Vaid do not teach features in the present application, Fries also fails to teach or suggest the features. Although Fries shows a Topics Dictionary in FIGs. 14A-14C, it is not a dictionary of a directed set of concepts, with one concept identified as a maximal element. Instead as described in column 12, lines 8-13 of Fries, the Topics Dictionary is a set of topics with associated keywords.

Also, Fries does not teach chains connecting the maximal element to selected concepts, or the selection of chains to form a basis. First, Fries does not teach using basis chains to construct vectors measuring the concreteness of a concept's representation. Indeed, Fries cannot teach basis chains connecting each element to the maximal element in the directed set, as there is no directed set and no maximal element. Instead of using basis chains, Fries says on column 20, lines 60-61, that the goal vector is based on clue stream features. While these features can include a semantic bit, the semantic bit does not have anything to do with basis chains.

Fries also fails to teach creating a state vector for each selected concept, where each state vector includes a measure of how concretely the concept is represented in each chain in the basis. While the SVM might measure how concretely a concept is measured by a query,

the SVM does not suggest a way to measure how concretely a concept is represented in a basis chain, as Fries does not use basis chains.

Just as Conklin, Luciw, and Vaid do not teach a template that includes a subset of vectors in a topological vector space, Fries similarly fails to teach this claim feature. Indeed, a search for the word template in Fries results in no hits. And merely teaching vectors with theoretical semantic content does not teach vectors explicitly constructed based on the selection of chains from a directed set. Indeed, none of the references cited by the Examiner teach state vectors explicitly constructed using the selection of chains from a directed set.

The Examiner has argued that because at least one of the cited references teaches the use of a dictionary, vectors, templates and policies, that the claims in the current application are obvious. However, these are not the only features explicit in the claims. The claims also include a dictionary of concepts arranged as a directed set, with one concept being a maximal concept. State vectors of selected concepts, including measurements of how concretely the concepts are represented in the basis chains, are also included in the claims. Further, using the template to store vectors is not taught or suggested by any of the references. The claims include an impact summary that is created with a different subset of vectors as a content stream is monitored.

Furthermore, in responding to the Applicant's response to the previous Office Action, the Examiner has said that the Applicant cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. The Applicant does not believe that it has attacked references individually, but rather, the Applicant has pointed out features that none of the references teach or suggest, either individually, or in combination with other references. Regardless, the Examiner bears the burden of showing a motivation to combine the references; the Applicant believe no such motivation is shown, as certain features of the claims are not taught by any of the references.

As none of these features are taught or suggested by Conklin, Luciw, Vaid or Fries, the combination of Conklin, Luciw, Vaid, and Fries fails to make obvious claims 1, 7, and 13, and the Examiner has failed to make a prima facie case of obviousness. Therefore, claims 1, 7, and 13 are patentable under 35 U.S.C. § 103(a) over Conklin in view of Luciw, Vaid and Fries. As such, claims 1-5, 7-11, 13-16, and 21-32 are allowable.

Referring to claim 5, the invention is directed toward a method according to claim 1, wherein monitoring a content stream includes: monitoring a portion of the content stream on the computer network; constructing the impact summary including the second subset of the

vectors in the topological vector space from the portion of the content stream; and extrapolating how close the entire content stream is to the template using the impact summary and the template.

Referring to claim 11, the invention is directed toward a program according to claim 7, wherein the monitoring software includes: monitoring software to monitor a portion of the content stream on the computer network; construction software to construct the impact summary including the second subset of the vectors in the topological vector space from the portion of the content stream; and extrapolation software to extrapolate how close the entire content stream is to the template from the portion of the content stream using the impact summary and the template.

Referring to claim 16, the invention is directed toward an apparatus according to claim 13, wherein: the monitor is adapted to monitor only a portion of the content stream on the computer network and construct the impact summary including the second subset of the vectors in the topological vector space from the portion of the content stream; and the policy enforcer is adapted to extrapolate how close the entire content stream is to the template using the impact summary and the template.

The Examiner argues that the feature of extrapolation is well known in the art, and refers to Ramamurthy, without formally citing to Ramamurthy. In fact, the Examiner even states that Rammurthy is not relied upon. The Applicant would like to point out that in order for the Examiner to make a prima facie rejection using 35 U.S.C. § 103(a), the Examiner is required to cite art that teaches all claim features. As the Examiner has not cited a reference that teaches or suggests extrapolation of content, the Examiner has not made a prima facie rejection of claims 5, 11 and 16 under 35 U.S.C. § 103(a).

The present application uses a vector comparison in order to make an extrapolation about content. None of the cited references provide instruction for how extrapolation can be used to predict whether a content stream is within a threshold distance of a template. Indeed, none of the cited references teach extrapolation.

As claims 5, 11 and 16 include features not taught or suggested by Conklin, Luciw, Vaid, nor Fries, claims 5, 11 and 16 are patentable under 35 U.S.C. § 103(a) over Conklin, Luciw, Vaid, and Fries. Accordingly, claims 5, 11 and 16 are allowable.

Referring to claim 22, the invention is directed towards a method according to claim 21, wherein measuring a distance includes using a Hausdorff distance function to measure the distance between the impact summary and the template.

Referring to claim 24, the invention is directed towards a program according to claim 23, wherein the measurement software includes measurement software to use a Hausdorff distance function to measure the distance between the impact summary and the template.

Referring to claim 26, the invention is directed towards an apparatus according to claim 25, wherein the distance measurer includes a Hausdorff distance measurer to use a Hausdorff distance function to measure the distance between the impact summary and the template.

The Examiner has argued that because Fries teaches measuring distance between vectors, that it would be obvious to use the Hausdorff distance function to measure the distance between the impact summary and the template. The Applicant notes that in order to make a prima facie rejection using 35 U.S.C. § 103(a), the Examiner is required to cite prior art teaching each feature in the claims. But as Fries fails to teach or suggest using the Hausdorff distance function, the Applicant believes that the Examiner has not sufficiently demonstrated that these claims that use the Hausdorff distance function are obvious. The Hausdorff distance function does not measure distance between points like Euclidean distance, and is not interchangeable with Euclidean distance functions. Thus, the fact that Fries mentions distance does not make the use of the Hausdorff distance function obvious.

Because it is not obvious to use the Hausdorff distance function to measure the distance between the impact summary and the template, claims 22, 24, and 26 are not obvious over Conklin, in view of Luciw, Vaid and Fries. Therefore claims 22, 24, and 26 are patentable under 35 U.S.C. § 103(a) over Conklin in view of Luciw, Vaid and Fries. As such, claims 22, 24, and 26 are allowable.

DOUBLE PATENTING

Claims 1-5, 7-11, 13-16, and 21-32 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-26 of co-pending application no. 09/653,713 in view of Conklin, Luciw, Vaid, and Fries. Claims 1-5, 7-11, 13-16 and 21-32 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting over U.S. Patent Application No. 09/512,963 in view of Conklin, Luciw, Vaid, and Fries. As the double patenting rejections are provisional, no action will be taken at this time. However, on notice of allowance of one of the co-pending applications, the Applicant is open to filing a terminal disclaimer to avoid the double patenting rejection.

For the foregoing reasons, reconsideration and allowance of claims 1-5, 7-11, 13-16, and 21-32 of the application as amended is solicited. The Examiner is encouraged to telephone the undersigned at (503) 222-3613 if it appears that an interview would be helpful in advancing the case.

Respectfully submitted,

MARGER JOHNSON & McCOLLOM, P.C.

Ariel S. Rogson

Ariel S. Rogson
Reg. No. 43,054

MARGER JOHNSON & McCOLLOM, P.C.
210 SW Morrison Street, Suite 400
Portland, OR 97204
503-222-3613

Customer No. 45842

ELEMENTARY LINEAR ALGEBRA

Second Edition

STEWART VENIT & WAYNE BISHOP

California State University at Los Angeles

Prindle, Weber & Schmidt
Boston

independent *and* span the subspace are of particular importance in linear algebra. In this section we investigate the characteristics of such sets.

BASIS FOR A SUBSPACE

DEFINITION

Let $\mathbf{S}$ be a subspace of $\mathbf{R}^m$. A set $\mathcal{S}$ of vectors in $\mathbf{S}$ is a **basis** for $\mathbf{S}$ if

i. $\mathcal{S}$ is linearly independent and

ii. $\mathcal{S}$ spans $\mathbf{S}$.

EXAMPLE 4.16

Show that the set of elementary vectors in $\mathbf{R}^m$, $\mathcal{S} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$, is a basis for $\mathbf{R}^m$.

By Exercise 29 of Section 4.1, $\mathcal{S}$ is linearly independent. Moreover $\mathcal{S}$ spans $\mathbf{R}^m$ (Example 4.13). Hence $\mathcal{S}$ is a basis for $\mathbf{R}^m$.

NOTE: The basis, $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$ is called the **standard basis** for $\mathbf{R}^m$.

EXAMPLE 4.17

Show that the set $\mathcal{S} = \{(1, 0, 0), (1, 2, 0), (1, 2, 3)\}$ is a basis for $\mathbf{R}^3$.

We can show that $\mathcal{S}$ is linearly independent and that $\mathcal{S}$ spans $\mathbf{R}^3$ at the same time. Let $\mathbf{w} = (w_1, w_2, w_3)$ be an arbitrary vector in $\mathbf{R}^3$. Then $\mathcal{S}$ spans $\mathbf{R}^3$ if there exist scalars c_1, c_2, c_3 such that

$$c_1(1, 0, 0) + c_2(1, 2, 0) + c_3(1, 2, 3) = (w_1, w_2, w_3).$$

Performing the scalar multiplications and equating corresponding components yields the linear system

$$\begin{aligned} c_1 + c_2 + c_3 &= w_1 \\ 2c_2 + 2c_3 &= w_2 \\ 3c_3 &= w_3, \end{aligned}$$

which has the coefficient matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}.$$

By Theorem 5 of Section 2.5 there is a unique solution to this system if the row-reduced echelon form of A is the identity matrix I. Performing the row-reduction we see that this is indeed the case, so $\mathcal{S}$ spans $\mathbf{R}^3$. Moreover, in doing the row-reduction and obtaining the identity matrix, we have also shown that $\mathcal{S}$ is linearly independent (Theorem 4 of Section 4.1). Consequently $\mathcal{S}$ is a basis for $\mathbf{R}^3$.

Now suppose that we wish to find a basis for the subspace $\mathbf{S}$ generated by a set, $\mathcal{S}$, of vectors in $\mathbf{S}$. By definition $\mathcal{S}$ spans $\mathbf{S}$, but $\mathcal{S}$ is not necessarily linearly independent. The following theorem and accompanying procedure show us how to *reduce a spanning set to a basis.*

IEERING SERIES

:lation

I LEVY

rogramming

GRAPH ALGORITHMS

SHIMON EVEN
Technion, Israel Institute of Technology

Computer Science Press

Chapter 2

TREES

2.1 TREE DEFINITIONS

Let $G(V, E)$ be an (undirected), finite or infinite graph. We say that G is *circuit-free* if there are no simple circuits in G. G is called a *tree* if it is connected and circuit-free.

Theorem 2.1: The following four conditions are equivalent:

(a) G is a tree.
(b) G is circuit-free, but if any new edge is added to G, a circuit is formed.
(c) G contains no self-loops and for every two vertices there is a unique simple path connecting them.
(d) G is connected, but if any edge is deleted from G, the connectivity of G is interrupted.

Proof: We shall prove that conditions (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d) $\Rightarrow$ (a).

(a) $\Rightarrow$ (b): We assume that G is connected and circuit-free. Let e be a new edge, that is $e \notin E$; the two endpoints of e, a and b, are elements of V. If $a = b$, then e forms a self-loop and therefore a circuit exists. If $a \neq b$, there is a path in G (without e) between a and b; if we add e, this path with e forms a circuit.

(b) $\Rightarrow$ (c): We assume that G is circuit-free and that no edge can be added to G without creating a circuit. Let a and b be any two vertices of G. If there is no path between them, then we can add an edge between a and b without creating a circuit. Thus, G must be connected. Moreover, if there are two simple paths, P and P', between a and b, then there is a circuit in G. To see this, assume that $P = e_1, e_2, \ldots, e_l$ and $P' = e_1', e_2', \ldots, e_m'$. Since both paths are simple, one cannot be the beginning of the other. Let i be the first index for which $e_i \neq e_i'$, and let v be the first vertex on $e_i, e_{i+1}, \ldots, e_l$ which is also on $e_i', e_{i+1}', \ldots, e_m'$. The two

2.4 DIRECTED TREE DEFINITIONS

A digraph $G(V, E)$ is said to have a *root* r if $r \in V$ and every vertex $v \in V$ is *reachable* from r; i.e., there is a directed path which starts in r and ends in v.

A digraph (finite or infinite) is called a *directed tree* if it has a root and its underlying undirected graph is a tree.

Theorem 2.5: Assume G is a digraph. The following five conditions are equivalent:

(a) G is a directed tree.
(b) G has a root from which there is a unique directed path to every vertex.
(c) G has a root r for which $d_{in}(r) = 0$ and for every other vertex v, $d_{in}(v) = 1$.
(d) G has a root and the deletion of any edge (but no vertices) interrupts this condition.
(e) The underlying undirected graph of G is connected and G has one vertex r for which $d_{in}(r) = 0$, while for every other vertex v, $d_{in}(v) = 1$.

Proof: We prove that (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d) $\Rightarrow$ (e) $\Rightarrow$ (a).

(a) $\Rightarrow$ (b): We assume that G has a root, say r, and its underlying undirected graph G' is a tree. Thus, by Theorem 2.1, part (c), there is a unique simple path from r to every vertex in G'; also, G' is circuit-free. Thus, a directed path from r to a vertex v, in G, must be simple and unique.

(b) $\Rightarrow$ (c): Here we assume that G has a root, say r, and a unique directed path from it to every vertex v. First, let us show that $d_{in}(r) = 0$. Assume there is an edge $u \xrightarrow{e} r$. There is a directed path from r to u, and it can be continued, via e, back to r. Thus, in addition to the empty path from r to itself (containing no edges), there is one more, in contradiction of the assumption of the path uniqueness. Now, we have to show that if $v \neq r$ then $d_{in}(v) = 1$. Clearly, $d_{in}(v) > 0$ for it must be reachable from r. If $d_{in}(v) > 1$, then there are at least two edges, say $v_1 \xrightarrow{e_1} v$ and $v_2 \xrightarrow{e_2} v$. Since there is a directed path P_1 from r to v_1, and a directed path P_2 from r to v_2, by adding e_1 to P_1 and e_2 to P_2 we get two different paths from r to v. (This proof is valid even if $v_1 = v_2$.)

(c) $\Rightarrow$ (d): This proof is trivial, for the deletion on any edge $u \xrightarrow{e} v$ will make v unreachable from r.